UNITED STATES PATENT AND TRADEMARK OFFICE

12

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/624,964	07/21/2003	Margaret Ann Bernal	SVL920030006US1	6263

47069 7590 08/11/2006

KONRAD RAYNES & VICTOR, LLP
ATTN: IBM54
315 SOUTH BEVERLY DRIVE, SUITE 210
BEVERLY HILLS, CA 90212

EXAMINER
SAEED, USMAAN

ART UNIT	PAPER NUMBER
2166	

DATE MAILED: 08/11/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/624,964	**Applicant(s)** BERNAL ET AL.	
	Examiner Usmaan Saeed	**Art Unit** 2166	

-- ***The MAILING DATE of this communication appears on the cover sheet with the correspondence address*** --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 May 2006*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-34* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-34* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *21 July 2003* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

Response to Amendment

1. Receipt of Applicant's Amendment, filed on 5/15/2006 is acknowledged. Claims 5, 7, 17, 19, 27, and 29 has been amended.

Specification

2. The amended specification was received on 5/15/2006 and is acceptable.

Drawings

3. The amended specification received on 5/15/2006 overcomes the drawing rejections and is acceptable.

Claim Objections

4. The amended claims 8 and 19 received on 5/15/2006 overcome the claim objections and are acceptable.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claims 1, 5-13, 17-23, and 27-34 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Bonner et al.** (**Bonner** hereinafter) U.S. PG Pub No. 2002/0029211 in view of **Bastawala et al**. (**Bastawala** hereinafter) U.S. Patent No 6,973,457.

With respect to claim 1, **Bonner** teaches **a method for making data available to an application program, comprising:**

"generating a cursor positioned with respect to a result table" as when a cursor is opened or initialized, the current row position of the cursor is the first record in the result table (**Bonner** Paragraph 0006). **"wherein the cursor specifies a search criteria"** as a static cursor does not display new rows inserted in the database after the cursor was opened, even if they match the search conditions of the cursor SELECT statement (**Bonner** Paragraph 0014). **"wherein the result table includes rows from a base table that satisfy the search criteria"** as the result table may be implemented in a work file or comprise the rows pointed to by the cursor in the base table (**Bonner** Paragraph 0007).

"receiving a fetch request indicating to position the cursor on a plurality of rows of the result table" as the application program may then issue fetch commands to move the current row position and fetch forward or backward by one or more rows or from the first or last row by one or more rows (**Bonner** Paragraph 0006). After the result table 50 is populated with rows from the database table according to any qualification criteria in the SELECT statement, the application 2 may issue FETCH statements to fetch rows of data from the result table or positioned UPDATE and DELETE commands to modify rows in the result table 50 (**Bonner** Paragraph 0079).

"positioning the cursor on the plurality of rows of the result table indicated in the fetch request that satisfy the search criteria" as control begins at block 250 with the executive 6 receiving the FETCH command. If (at block 252) the FETCH is "insensitive", then the database program 6 would position (at block 254) the cursor to the position specified in the FETCH operation, e.g., PRIOR, FIRST, LAST, CURRENT, etc. and then return (at block 256) the row at the new cursor position in the result table 50 (**Bonner** Paragraph 0089).

Bonner discloses the elements of claim 1 as noted above but does not explicitly teach the step of having **"cursor on a plurality of rows of the result table."**

However, **Bastawala** discloses **"cursor on a plurality of rows of the result table"** as when the user at client 100 accesses the result set, the current position of the cursor points at a row or set of rows in the result set (**Bastawala** Col 3, Lines 24-26).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Bastawala's** teaching would have allowed **Bonner** to retrieve either a row at a time or in groups (**Bastawala** Col 1, Lines 35-40), thus providing a mechanism for accessing the records more efficiently.

Claims 13 and 23 are essentially the same as claim 1 except they set forth the claimed invention as a system and an article of manufacture, and are rejected for the same reasons as applied hereinabove.

With respect to claim 5, **Bonner** teaches **"the method of claim 1, wherein the cursor is positioned on a current plurality of rows of the result table before receiving the fetch request, and wherein positioning the cursor further comprises"** as when a cursor is opened or initialized, the current row position of the cursor is the first record in the result table (**Bonner** Paragraph 0006).

"positioning the cursor on a number of rows with respect to one row of the result table having rows" as control begins at block 250 with the executive 6 receiving the FETCH command. If (at block 252) the FETCH is "insensitive", then the database program 6 would position (at block 254) the cursor to the position specified in the FETCH operation, e.g., PRIOR, FIRST, LAST, CURRENT, etc. and then return (at block 256) the row at the new cursor position in the result table 50 (**Bonner** Paragraph 0089). **"satisfy the search criteria"** as a static cursor does not display new rows inserted in the database after the cursor was opened, even if they match the search conditions of the cursor SELECT statement (**Bonner** Paragraph 0014).

Bonner discloses the elements of claim 5 as noted above but does not explicitly teach the step of **"determining/comprising a rowset size" and "cursor on a plurality of rows of the result table."**

However, **Bastawala** discloses, **"determining/comprising a rowset size"** as within the scrollbar is a tab that indicates the relative position of the display within the entire result set. In addition, the size of the tab is often configured to reflect the relative size of the display window compared to the entire result set. In conventional systems, two separate operations are typically performed to obtain the relevant information needed to size and position a scrollbar tab. A first SQL statement is used to obtain the size (number of rows) of the result set (e.g., "count *") (**Bastawala** Col 6, Lines 59-67 and Col 6, Lines 1-10) **and "cursor on a plurality of rows of the result table"** as when the user at client 100 accesses the result set, the current position of the cursor points at a row or set of rows in the result set (**Bastawala** Col 3, Lines 24-26).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because

Bastawala's teaching would have allowed **Bonner** to retrieve either a row at a time or in groups (**Bastawala** Col 1, Lines 35-40), thus providing a mechanism for accessing the records more efficiently. The rowset size would have allowed to reflect the relative size of the display window compared to the entire result set (**Bastawala** Col 6, Lines 59-67).

Claims 17 and 27 are essentially the same as claim 5 except they set forth the claimed invention as a system and an article of manufacture, and are rejected for the same reasons as applied hereinabove.

With respect to claim 6, **Bonner** teaches **the method of claim 5, wherein positioning the cursor on the number of rows with respect to one row of the result table comprises one of:**

"positioning the cursor on a number of rows preceding a first row of the current plurality of rows" as BEFORE, which positions the cursor before the first row of the result table 50 (**Bonner** Paragraph 0085). **"satisfy the search criteria"** as a static cursor does not display new rows inserted in the database after the cursor was opened, even if they match the search conditions of the cursor SELECT statement (**Bonner** Paragraph 0014).

"positioning the cursor on a number of rows from a first row of the result table" as FIRST, which positions the cursor on the first row of the result table 50 and fetches the row (**Bonner** Paragraph 0082). **"satisfy the search criteria"** as a static cursor does not display new rows inserted in the database after the cursor was opened, even if they match the search conditions of the cursor SELECT statement (**Bonner** Paragraph 0014).

"positioning the cursor on a number of rows preceding an end of the result table" as LAST, which positions the cursor on the last row of the result table 50 and fetches the row (**Bonner** Paragraph 0083). **"satisfy the search criteria"** as a static cursor does not display new rows inserted in the database after the cursor was opened, even if they match the search conditions of the cursor SELECT statement (**Bonner** Paragraph 0014).

Bonner discloses the elements of claim 6 as noted above but does not explicitly teach the step of having **"cursor on a plurality of rows of the result table."**

However, **Bastawala** discloses **"cursor on a plurality of rows of the result table"** as when the user at client 100 accesses the result set, the current position of the cursor points at a row or set of rows in the result set (**Bastawala** Col 3, Lines 24-26).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Bastawala's** teaching would have allowed **Bonner** to retrieve either a row at a time or in groups (**Bastawala** Col 1, Lines 35-40), thus providing a mechanism for accessing the records more efficiently.

Claims 18 and 28 are essentially the same as claim 6 except they set forth the claimed invention as a system and an article of manufacture, and are rejected for the same reasons as applied hereinabove.

With respect to claim 7, **Bonner** teaches **"the method of claim 1, wherein the cursor is positioned on a current plurality of rows of the result table before receiving the fetch request specifying an integer k, and wherein positioning the cursor further comprises"** as FIG. 12 illustrates logic implemented in the database program to FETCH ABSOLUTE from a current position in the result table to a kth position in the result table in accordance with preferred embodiments of the present invention (**Bonner** Paragraph 0035).

"positioning the cursor on a number of rows that satisfy the search criteria" as after the result table 50 is populated with rows from the database table according to any qualification criteria in the SELECT statement, the application 2 may issue FETCH statements to fetch rows of data from the result table or positioned UPDATE and DELETE commands to modify rows in the result table 50 (**Bonner** Paragraph 0079). **"and is positioned with respect to k rows from row of the result table having rows that satisfy the search criteria"** as FIG. 12 illustrates logic implemented in the database program to FETCH ABSOLUTE from a current position in the result table to a kth position in the result table in accordance with preferred embodiments of the present invention (**Bonner** Paragraph 0035).

Bonner discloses the elements of claim 7 as noted above but does not explicitly teach the step of **"determining/comprising a rowset size" and "cursor on a plurality of rows of the result table."**

However, **Bastawala** discloses, **"determining/comprising a rowset size"** as within the scrollbar is a tab that indicates the relative position of the display within the entire result set. In addition, the size of the tab is often configured to reflect the relative size of the display window compared to the entire result set. In conventional systems, two separate operations are typically performed to obtain the relevant information needed to size and position a scrollbar tab. A first SQL statement is used to obtain the size (number of rows) of the result set (e.g., "count *") (**Bastawala** Col 6, Lines 59-67 and Col 6, Lines 1-10) **and "cursor on a plurality of rows of the result table"** as when the user at client 100 accesses the result set, the current position of the cursor points at a row or set of rows in the result set (**Bastawala** Col 3, Lines 24-26).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Bastawala's** teaching would have allowed **Bonner** to retrieve either a row at a time or in groups (**Bastawala** Col 1, Lines 35-40), thus providing a mechanism for accessing the records more efficiently. The rowset size would have allowed to reflect the relative size of the display window compared to the entire result set (**Bastawala** Col 6, Lines 59-67).

Claims 19 and 29 are essentially the same as claim 7 except they set forth the claimed invention as a system and an article of manufacture, and are rejected for the same reasons as applied hereinabove.

With respect to claim 8, **Bonner** teaches **the method of claim 7, wherein positioning the cursor on a number of rows that satisfy the search criteria and is positioned with respect to k rows from row of the result table comprises one of:**

"positioning the cursor on a number of rows that satisfy the search criteria" as after the result table 50 is populated with rows from the database table according to any qualification criteria in the SELECT statement, the application 2 may issue FETCH statements to fetch rows of data from the result table or positioned UPDATE and DELETE commands to modify rows in the result table 50 (**Bonner** Paragraph 0079). **"and precede k rows preceding a first row of the current plurality of rows that satisfy the search criteria"** as ABSOLUTE: Evaluates the host variable or integer constant to an integral value k, and then moves the cursor position to the kth row in the result table 50 if k>0 or to k rows from the bottom of the table if k<0 (**Bonner** Paragraph 0087).

"positioning the cursor on a number of rows that satisfy the search criteria" as after the result table 50 is populated with rows from the database table according to any qualification criteria in the SELECT statement, the application 2 may issue FETCH statements to fetch rows of data from the result table or positioned UPDATE and DELETE commands to modify rows in the result table 50 (**Bonner**

Paragraph 0079). **"and follow a number of rows equal to the rowset size from a kth row from a first row of the result table"** as the data manager 16 executes the FETCH ABSOLUTE by positioning (at block 410) the cursor pointer to the first entry in the result table 50 and then moving the cursor downward to the kth entry from the top in the result table 50 (**Bonner** Paragraph 0103).

"positioning the cursor on a number of rows that satisfy the search criteria" as after the result table 50 is populated with rows from the database table according to any qualification criteria in the SELECT statement, the application 2 may issue FETCH statements to fetch rows of data from the result table or positioned UPDATE and DELETE commands to modify rows in the result table 50 (**Bonner** Paragraph 0079). **"and precedes k rows that satisfy the search criteria preceding a last row of the result table"** as ABSOLUTE: Evaluates the host variable or integer constant to an integral value k, and then moves the cursor position to the kth row in the result table 50 if k>0 or to k rows from the bottom of the table if k<0 (**Bonner** Paragraph 0087).

Bonner discloses the elements of claim 8 as noted above but does not explicitly teach the step of having **"cursor on a plurality of rows of the result table."**

However, **Bastawala** discloses **"cursor on a plurality of rows of the result table"** as when the user at client 100 accesses the result set, the current position of the cursor points at a row or set of rows in the result set (**Bastawala** Col 3, Lines 24-26).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because

Bastawala's teaching would have allowed **Bonner** to retrieve either a row at a time or in groups (**Bastawala** Col 1, Lines 35-40), thus providing a mechanism for accessing the records more efficiently.

Claims 20 and 30 are essentially the same as claim 8 except they set forth the claimed invention as a system and an article of manufacture, and are rejected for the same reasons as applied hereinabove.

With respect to claim 9, **Bonner** teaches **the method of claim 1, further comprising:**

"receiving a request to modify at least one row in the rows on which the cursor is positioned" as after the result table 50 is populated with rows from the database table according to any qualification criteria in the SELECT statement, the application 2 may issue FETCH statements to fetch rows of data from the result table or positioned UPDATE and DELETE commands to modify rows in the result table 50 (**Bonner** Paragraph 0079).

"modifying the at least one row on which the cursor is positioned as indicated in the request" as after the result table 50 is populated with rows from the database table according to any qualification criteria in the SELECT statement, the application 2 may issue FETCH statements to fetch rows of data from the result table or positioned UPDATE and DELETE commands to modify rows in the result table 50

(**Bonner** Paragraph 0079). Further, the data manager 16 would also perform any updates or modifications to rows in the database tables (**Bonner** Paragraph 0041).

Claims 21 and 31 are essentially the same as claim 9 except they set forth the claimed invention as a system and an article of manufacture, and are rejected for the same reasons as applied hereinabove.

With respect to claim 10, **Bonner** teaches **"the method of claim 9, wherein the modification comprises updating or deleting the at least one row on which the cursor is positioned as indicated in the request"** as after the result table 50 is populated with rows from the database table according to any qualification criteria in the SELECT statement, the application 2 may issue FETCH statements to fetch rows of data from the result table or positioned UPDATE and DELETE commands to modify rows in the result table 50 (**Bonner** Paragraph 0079).

Claim 32 is essentially the same as claim 10 except it sets forth the claimed invention as an article of manufacture, and is rejected for the same reasons as applied hereinabove.

With respect to claim 11, **Bonner** teaches **"the method of claim 1, wherein the cursor comprises one of a static cursor or dynamic cursor, wherein if the cursor is static, then the cursor is either sensitive or insensitive to changes in the base**

table from which the result table is generated" as cursors may be categorized as forward-only or scrollable. If the cursor is scrollable then they can be either static, keyset or dynamic (**Bonner** Paragraph 0016). FIG. 4 illustrates a format of a DECLARE cursor command in accordance with the preferred embodiments. To declare a scrollable cursor, the application 2 must specify either "insensitive" or "sensitive static" (**Bonner** Paragraph 0045).

Claims 22 and 33 are essentially the same as claim 11 except they set forth the claimed invention as a system and an article of manufacture, and are rejected for the same reasons as applied hereinabove.

With respect to claim 12, **Bonner** does not explicitly teach **"the method of claim 1, wherein the cursor is positioned on a current plurality of rows of the result table before receiving the fetch request, and wherein the current plurality of rows is a different number than a number of the rows on which the cursor is positioned in response to the fetch request."**

However, **Bastawala** discloses **"the method of claim 1, wherein the cursor is positioned on a current plurality of rows of the result table before receiving the fetch request, and wherein the current plurality of rows is a different number than a number of the rows on which the cursor is positioned in response to the fetch request"** as FIGS. 5a and 5b depict a fourth example scenario for scrolling a cursor according to an embodiment of the invention. FIG. 5a shows a display window 502 that is presently displaying rows 16 to 18 in a query result set. Based upon the present cursor position, the client cache 508 presently contains rows 16 to 18 of the result set as shown in FIG. 5b. Consider if the user decides to scroll and resize the display window, and hence scroll the cursor, such that rows 15 to 19 of the result set are displayed (as shown in modified display window 502'). Here, it appears that we are scrolling in both the forward and backward directions at the same time (**Bastawala** Col 5, Lines 57-67).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Bastawala's** teaching would have allowed **Bonner** to retrieve the row only in the display range/(pointed by cursor) in order to transmit less amount of data over the network from a server to the client (**Bastawala** Col 6, Lines 26-32).

Claim 34 is essentially the same as claim 12 except it sets forth the claimed invention as an article of manufacture, and is rejected for the same reasons as applied hereinabove.

6. Claims 2-4, 14-16, and 24-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Bonner et al.** U.S. PG Pub No. 2002/0029211, in view of **Bastawala et al.** U.S. Patent No 6,973,457 as applied to claims 1, 5-13, 17-23, and 27-34 above, further in view of **Vicik et al.** (**Vicik** hereinafter) (U.S. Patent No. 5,835,904).

With respect to claim 2, **Bonner and Bastawala** do not explicitly teach **"the method of claim 1, further comprising: placing a lock on the plurality of rows of the result table on which the cursor is positioned."**

However, **Vicik** discloses **"the method of claim 1, further comprising: placing a lock on the plurality of rows of the result table on which the cursor is positioned"** as for example, cursors are commonly used to set "lock conditions", i.e. to control access to certain data (**Vicik** Col 2, Lines 35-38). Moreover, the client lock setting on data within a result set also affected the ability of that client or other clients to scroll through or update rows in this result set, particularly in cases where this result set spanned multiple tables (**Vicik** Col 7, Lines 13-17).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Vicik's** teaching would have allowed **Bonner and Bastawala** to provide for increased control over cursor lock options, and an ability to share cursors and search plan across multiple nodes in a network and an increase in the overall operating efficiency of a database application (**Vicik** Abstract).

Claims 14 and 24 are essentially the same as claim 2 except they set forth the claimed invention as a system and an article of manufacture, and are rejected for the same reasons as applied hereinabove.

With respect to claim 3, **Bonner** does not explicitly teach **"the method of claim 2, wherein the fetch request is received from a client at a server, further comprising: returning, by the server, the plurality of rows at the server on which the cursor is positioned to the client that sent the fetch request, wherein the lock is placed on the plurality of rows at the server to block the plurality of rows on which the cursor is positioned."**

However, **Bastawala** teaches, **"the fetch request is received from a client at a server"** as FIG. 1a shows an architecture for implementing scrollable cursors according to one embodiment of the invention. In the system shown in FIG. 1a, a user at a client computer 100 initiates a SQL query that is executed at a server 102. The server 102 accesses a database 110 to retrieve data responsive to the SQL query. A server cache 106 stores a result set for the query (**Bastawala** Col 2, 53-59). **"returning, by the server, the plurality of rows at the server on which the cursor is positioned to the client that sent the fetch request"** as therefore, rows 15 to 19 must be fetched from the server before the cursor is scrolled to the appropriate position in the result set. Since the entire result set is already cached at the server cache 212, the specified rows 214 can be immediately sent from the server to the client without regenerating the data (**Bastawala** Col 5, Lines 5-10).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Bastawala's** teaching would have allowed **Bonner** to retrieve the row only in the display range/(pointed by cursor) in order to transmit less amount of data over the network from a server to the client (**Bastawala** Col 6, Lines 26-32).

Bonner and Bastawala disclose the elements of claim 3 as noted above, but fail to teach the step of **"the lock is placed on the plurality of rows at the server to block the plurality of rows on which the cursor is positioned."**

However, **Vicik** discloses **"the lock is placed on the plurality of rows at the server to block the plurality of rows on which the cursor is positioned"** as FIG. 4 is a table that graphically sets forth the concurrency, lock options and their effect on the cursor owner for database cursors implemented at the server in accordance with a preferred embodiment of the invention (**Vicik** Col 4, Lines 51-53).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Vicik's** teaching would have allowed **Bonner and Bastawala** to provide for increased control over cursor lock options, and an ability to share cursors and search plan across multiple nodes in a network and an increase in the overall operating efficiency of a database application (**Vicik** Abstract).

Claims 15 and 25 are essentially the same as claim 3 except they set forth the claimed invention as a system and an article of manufacture, and are rejected for the same reasons as applied hereinabove.

With respect to claim 4, **Bonner** teaches **the method of claim 2, further comprising:**

"receiving a subsequent fetch request to reposition the cursor on at least one row of the result table" as if the FETCH is SENSITIVE, then the data manager 16 repositions (at block 258) the cursor (current row pointer) to the row in the result table 50 according to the operation specified in the FETCH statement, e.g., next, prior, first, k rows forward or backward in a relative or absolute operation, etc (**Bonner** Paragraph 0090).

Bonner and Bastawala disclose the elements of claim 4 as noted above, but fail to teach the step of **"releasing the lock on the plurality of rows of the result table on which the cursor is currently positioned before being repositioned."**

However, **Vicik** discloses, **"releasing the lock on the plurality of rows of the result table on which the cursor is currently positioned before being repositioned"** as the client isolates or locks data at the server by issuing fictional updates and withholding a commit instruction to release the lock until the cursor moved from the data (**Vicik** Col 9, Lines 37-41). Therefore the reference has instructions, which releases the lock on the data/rows.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Vicik's** teaching would have allowed **Bonner and Bastawala** to provide for increased control over cursor lock options, and an ability to share cursors and search plan across multiple nodes in a network and an increase in the overall operating efficiency of a database application (**Vicik** Abstract).

Claims 16 and 26 are essentially the same as claim 4 except they set forth the claimed invention as a system and an article of manufacture, and are rejected for the same reasons as applied hereinabove.

Response to Arguments

7. Applicant's arguments filed on 5-15-2006 have been fully considered but they are not persuasive.

Regarding claim 1, 13 and 23 applicant argues that **Bonner and Bastawala** do not teach **"positioning the cursor on a plurality of rows of the result table."**

In response to the preceding argument, Examiner respectfully submits that **Bastawala** teaches, **"positioning the cursor on a plurality of rows of the result table"** as the current position of the cursor points at a row or set of rows in the result set (**Bastawala** Col 3, Lines 24-26). The current position of the cursor pointing at a row or set of rows in the result set is one of the embodiments of Bastawala.

Regarding claims 5, 7, 17, 19, 27, and 29 applicant argues **"nowhere is there any teaching or suggestion of the claim requirement of a rowset size indicating the number of rows on which the cursor is positioned. Instead, the size of the display window is the number of rows displayed, not the number of rows on which a cursor is positioned."**

In response to the preceding argument, Examiner respectfully submits that **Bastawala** teaches **"determining/comprising a rowset size"** as within the scrollbar is a tab that indicates the relative position of the display within the entire result set. In addition, the size of the tab is often configured to reflect the relative size of the display window compared to the entire result set. In conventional systems, two separate operations are typically performed to obtain the relevant information needed to size and position a scrollbar tab. A first SQL statement is used to obtain the size (number of rows) of the result set (e.g., "count *") (Bastawala Col 6, Lines 59-67).

Further **Bastawala** teaches the display range 210 corresponding to the new cursor position (rows 15 to 19) falls outside the set of rows that are locally cached in client cache 208. Therefore, rows 15 to 19 must be fetched from the server before the cursor is scrolled to the appropriate position in the result set. Since the entire result set is already cached at the server cache 212, the specified rows 214 can be immediately sent from the server to the client without regenerating the data (**Bastawala** Col 6, Lines 1-10).

Regarding claims 2, 14, and 24 applicant argues nowhere cited references disclose **"the requirement of placing a lock on a plurality of rows on which the cursor is positioned."**

In response to the preceding argument, Examiner respectfully submits that **Vicik** teaches **"placing a lock on the plurality of rows of the result table on which the cursor is positioned"** as for example, cursors are commonly used to set "lock conditions", i.e. to control access to certain data (**Vicik** Col 2, Lines 35-38). Moreover, the client lock setting on data within a result set also affected the ability of that client or other clients to scroll through or update rows in this result set, particularly in cases where this result set spanned multiple tables (**Vicik** Col 7, Lines 13-17).

Further claim 2 depends from claim 1 and in claim 1 **Bastawala** teaches **"positioning the cursor on a plurality of rows of the result table"** as the current position of the cursor points at a row or set of rows in the result set (**Bastawala** Col 3, Lines 24-26). The current position of the cursor pointing at a row or set of rows in the result set is one of the embodiments of Bastawala.

Regarding claims 3, 15, and 25 applicant argues nowhere cited references teach that **"the returned rows are those on which cursor is positioned."**

In response to the preceding argument, Examiner respectfully submits that **Bastawala** teaches **"the returned rows are those on which cursor is positioned"** as therefore, rows 15 to 19 must be fetched from the server before the cursor is scrolled to the appropriate position in the result set. Since the entire result set is already cached at the server cache 212, the specified rows 214 can be immediately sent from the server to the client without regenerating the data (**Bastawala** Col 5, Lines 5-10). The partial result set cached at the client computer is first checked to see if requested data is present. If so, then the requested data is fetched from the client cache and the current position of the cursor is moved to the appropriate position in the result set (**Bastawala** Abstract).

Further claim 3 depends from claim 2, which depends from claim 1, and in claim 1 **Bastawala** teaches **"positioning the cursor on a plurality of rows of the result table"** as the current position of the cursor points at a row or set of rows in the result set (**Bastawala** Col 3, Lines 24-26).

Further **Bonner** discloses an insensitive fetch command returns the row from the result table 50 without accessing the corresponding row in the base table 60 (**Bonner** Paragraph 0079).

Conclusion

8. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within

TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Contact Information

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Usmaan Saeed whose telephone number is (571)272-4046. The examiner can normally be reached on M-F 8-5.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hosain Alam can be reached on (571)272-3978. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Usmaan Saeed
Patent Examiner
Art Unit: 2166

Leslie Wong
Primary Examiner

US
August 01, 2006